AMENDMENTS TO THE SPECIFICATION

-Please amend the specification at the paragraph beginning at page 5, line 23 as follows:

With reference to the drawings, FIG. 1 shows a FDTD algorithm when using the internal memory organization scheme or system 10 in accordance with an embodiment of the present invention. Specifically, FIG. 1 shows the routing required to satisfy all data dependencies required by computation engines ~~12~~ 12-1 to 12-6. Most of the other labeled blocks in FIG. 1 are the internal memory banks. Scheme 10 includes a plurality of input memory banks 14 that connect to corresponding one-cycle delay elements 16. Delay elements 16, in turn, connect to computation engines ~~12~~ 12-1 to 12-6. Another plurality of input memory banks ~~18~~ 18-1 to 18-6 connect to computation engines ~~12~~ 12-1 to 12-6, via corresponding one-cycle delay elements ~~16~~ 20-1 to 20-6. Each computation engine ~~12~~ 12-1 to 12-6 connects to a corresponding output memory bank ~~22~~ 22-1 to 22-6.

-Please amend the specification at the paragraph beginning at page 6, line 14 as follows:

Memory banks ~~14~~ 14-1 to 14-11, ~~18~~ 18-1 to 18-6, ~~22~~ 22-1 to 22-6 represent cache memory, which is very fast random access memory (RAM). Typically, this is built into chips, allowing for much faster access than external DRAM or SRAM. The tradeoff is that it is much smaller, meaning that data can only reside there (i.e., it is "cached") for short periods of time before being placed back into larger storage to make room for newer data. As a microprocessor processes data, it looks first in the cache memory and if it finds data there (from a previous reading of data), it uses that data rather than going to the slower main memory to find it. When data is not found, this is commonly known as a "cache miss", and is very expensive in terms of performance. There is a related phenomenon known as "cache collision" or "thrashing" where certain data competes with other data for a certain location in cache, forcing the processor to repeatedly swap the competing pieces to and from slower memory. The present invention eliminates both of these problems by "prefetching" all required data without conflicts (see blocks

2

504 and 506 of FIG. 5, discussed below).

-Please amend the specification at the paragraph beginning at page 7, line 3 as follows:

Delay elements 16, 20-1 to 20-6 include a circuit that produces an output waveform similar to its input waveform, only delayed by a certain amount of time. Delay elements 16, 20-1 to 20-6 may thus include flip-flop chains (or shift registers), transmission gate based delay elements, cascaded inverter based delay elements, voltage-controlled based delay elements, etc. FIG. 2 shows an enlarged view of one of input memory banks 14 of the internal memory organization scheme 10. The smaller elements represent one-cycle delay elements 16, 20-1 to 20-6. Since each input memory bank 14 14-2 to 14-11 is dual-ported, there are two separate data paths 26, 28 exiting each input memory bank 14 14-2 to 14-11 and entering delay elements 16.

-Please amend the specification at the paragraph beginning at page 7, line 10 as follows:

Computation engines 12 12-1 to 12-6 include logic circuitry that takes data from input memory banks 14 14-1 to 14-11, 18 18-1 to 18-6 and delay elements 16, 20-1 to 20-6 perform performs FDTD calculations on this data, and outputs the results to output memory banks 22 22-1 to 22-6.

-Please amend the specification at the paragraph beginning at page 7, line 23 as follows:

Although the internal memory organization scheme 10 of FIG. 1 shows distinct numbers of memory banks 14 14-1 to 14-11, 18 18-1 to 18-6, 22 22-1 to 22-6, delay elements 16, 20-1 to 20-6 and computation engines 12 12-1 to 12-6, the present invention is not limited to these distinct numbers. Rather, the internal memory organization scheme 10 may include more or less memory banks 14 14-1 to 14-11, 18 18-1 to 18-6, 22 22-1 to 22-6, delay elements 16, and computation engines 12 12-1 to 12-6 than are shown in FIG. 1.

-Please amend the specification at the paragraph beginning at page 8, line 17 as follows:

The preferred embodiment of the present invention uses the internal, dual-port memory banks ~~14~~ 14-1 to 14-11 available in modern FPGA chips, but it may be built from other internal memory resources. The system 10 operates by fetching data from DRAM, which is then stored in internal cache and used in the FDTD calculations. The amount of data fetched at one time is referred to hereinafter as a "chunk" and contains all the values needed to perform a certain number of calculations. This is an efficient use of DRAM because data is highly collocated, meaning that it can be streamed effectively. To see the operation of this system, the data dependences in each of the three Cartesian directions are examined to show that they are satisfied.

-Please amend the specification at the paragraph beginning at page 9, line 10 as follows:

Data dependencies in the "j" direction cannot be handled in the same way as those in the "i" direction because the time delay would be much too long. Hundreds of cycles of delay could be required because the repetition interval would be based on row length, and this is expensive in terms of space requirements in FPGAs. An easy alternative to this is to use the second port of the internal dual-port memory banks ~~14~~ 14-1 to 14-11. This bank can be addressed independently, and can therefore provide any required data in this direction.

-Please amend the specification at the paragraph beginning at page 9, line 16 as follows:

Data dependencies in the "k" direction cannot be handled by either of the previous techniques. The time-delay technique is invalid for the same reasons it was invalid in the "j" direction, i.e., the delay would be too long. The easiest solution is to add extra memory banks ~~18~~ 18-1 to 18-6 for specially storing fields required by dependencies in the "k" direction.

-Please amend the specification at the paragraph beginning at page 9, line 20 as follows:

Applying all the techniques discussed above, the system or scheme 10 of the present invention includes: one memory bank ~~14~~ 14-1 to 14-11 for each field type, and one memory bank ~~18~~ 18-1 to 18-6 for any field that shows dependence in the "k" direction. One of the channels on each memory bank ~~14~~ 14-2 to 14-11 will have a delay element 16 attached thereto, allowing it to handle fields with no direction dependence as well as those with "i" dependencies, and the other channel on each memory bank ~~14~~ 14-1 to 14-11 will be used for "j" dependencies. One extra set of memory banks ~~22~~ 22-1 to 22-6 may be used to buffer updated fields before they are stored back to bulk memory.

-Please amend the specification at the paragraph beginning at page 10, line 16 as follows:

The method for using the internal memory organization scheme 10 of the present invention is shown generally in FIG. 5 as reference numeral 500. The method 500 starts at step 502 and includes a step 504 of loading dual fields for a chunk into the dual field memory banks ~~14~~ 14-1 to 14-11 (until the memory banks are full). While this is happening, the old values in these dual field memory banks ~~14~~ 14-1 to 14-11 are moved to memory banks that store k-dependent fields. A next step 506 includes loading primary fields for a chunk into the primary memory banks ~~18~~ 18-1 to 18-6 (this will be smaller than the dual fields by one row length). In step 508, computations begin and iterate over the primary fields. Step 510 includes store updated fields in the "updated" memory banks ~~22~~ 22-1 to 22-6 when updated fields emerge from computational hardware ~~12~~ 12-1 to 12-6. Step 512 includes writing back updated fields to bulk storage. Step 514 checks to see if method 500 is complete. If method 500 is complete, the process stops at step 516, otherwise the process moves to next chunk of data and repeats method 500.